

# **SWARM INTELLIGENCE: UMA ANÁLISE BIBLIOMÉTRICA<sup>1</sup>**

---

<sup>1</sup> Artigo apresentado como requisito parcial para a conclusão do curso de Graduação em Ciência da Computação da Universidade do Sul de Santa Catarina – UNISUL. 2020.

# SWARM INTELLIGENCE: UMA ANÁLISE BIBLIOMÉTRICA

Elias Dacoregio Heinzen<sup>II</sup>

---

<sup>II</sup> Acadêmico do curso de Ciência da Computação da Universidade do Sul de Santa Catarina – Unisul. E-mail: [elias.heinzen@unisul.br](mailto:elias.heinzen@unisul.br).

**Resumo:** Este artigo apresenta alguns dos métodos vistos na computação inspirada no comportamento natural de coletivos, sendo utilizados para a solução de problemas de alta complexidade computacional. As metaheurísticas abaixo apresentadas são a otimização por colônia de formigas e a otimização por enxame de partículas. Por último, dou destaque à aplicação prática de tais conceitos em uma área de grande interesse, a robótica de enxame.

**Palavras-chave:** Inteligência de Enxame, Inteligência Artificial, Computação Natural.

**Abstract:** This article presents some of the methods seen in computing inspired by the natural behavior of collectives, being used to solve problems of high computational complexity. The metaheuristics presented below are optimization by ant colony and optimization by particle swarm. Finally, I emphasize the practical application of such concepts in an area of great interest, swarm robotics.

**Keywords:** Swarm Intelligence, Artificial Intelligence, Natural Computing.

## 1 INTRODUÇÃO

Um dos ramos mais promissores da ciência da computação é a chamada Computação Natural (CN), que estuda, compreende e aplica padrões complexos, encontrados na natureza, a fim de resolver problemas e aperfeiçoar sistemas existentes. Alguns exemplos já bem conhecidos e estabelecidos são o velcro, o sonar e o submarino, baseados respectivamente nas plantas, nos morcegos e nos peixes.

Ainda dentro da computação natural, temos um subgrupo de estudos e aplicações chamado de Computação Inspirada na Natureza. Esta é a mais antiga e bem consolidada área dentro da CN, tendo como objetivo implementar modelos computacionais baseados em casos comuns observados na natureza, sendo possível então a implementação desses modelos computacionais para a solução de problemas e otimização de processos computacionais complexos.

Foi a partir da computação inspirada na natureza que surgiram as recentes áreas de Computação Evolucionária, Redes Neurais Artificiais, Sistemas Imunológicos Artificiais, assim como a Inteligência de Enxame, sendo o objetivo do presente artigo, o esclarecimento desta última.

---

<sup>III</sup> Mestre em Engenharia de Produção – Universidade Federal de Santa Catarina. Professor Titular na Universidade do Sul de Santa Catarina – UNISUL.

A Swarm Intelligence (em português, Inteligência de Enxame ou IE) pode ser definida como a disciplina que estuda o comportamento coletivo de sistemas que funcionam de forma auto-organizada e descentralizada, e tem como base os coletivos do mundo animal, como por exemplo, colônias de formigas e cupins, cardumes de peixes, bandos de pássaros e rebanhos de animais terrestres.

Os sistemas de IE normalmente são compostos por uma população de agentes simples (boids) que interagem localmente entre si e com o ambiente, apresentando comportamento emergente, isto é, padrões complexos formados a partir de uma multiplicidade de interações simples.

A Inteligência de Enxame apresenta capacidades de aprendizado não-supervisionado e resiliência aos extremos locais, características da computação cognitiva.

Pode ser utilizada para a criação de sistemas multi-robóticos, bem como para a solução, por aproximação, de problemas de otimização de alta complexidade (NP-completos) e também para a análise de grandes quantidades de dados.

## **2 METODOLOGIA**

Para o presente trabalho, foi definida a análise bibliométrica como a metodologia de pesquisa.

Através da análise de trabalhos científicos previamente publicados a respeito de determinado assunto, é possível fazer um panorama do atual estado do mesmo, bem como um melhor entendimento, de forma resumida, deste assunto.

Na pesquisa em questão, foi utilizado tanto o livro sobre o assunto (Swarm Intelligence, de BONABEU e DORIGO), quanto alguns artigos mais específicos, tratando a respeito de cada um dos algoritmos apresentados.

## **3 CONTEXTUALIZAÇÃO**

### **3.1 Ant Colony Optimization (ACO)**

A Otimização de Colônia de Formigas, ou Ant Colony Optimization (ACO) é uma metaheurística baseada na população que pode ser usada para encontrar soluções aproximadas para problemas de otimização difíceis (NP-completos).

No ACO, um conjunto de agentes de software chamados formigas artificiais busca boas soluções para um determinado problema de otimização. Para aplicar ACO, o problema de otimização é transformado no problema de encontrar o melhor caminho em um grafo ponderado. As formigas artificiais constroem soluções de forma incremental movendo-se no grafo.

O processo de construção da solução é estocástico e polarizado por um modelo de feromônio, ou seja, um conjunto de parâmetros associados aos componentes do grafo (nós ou arestas) cujos valores são modificados em tempo de execução pelas formigas.

### **3.1.1 Exemplo**

Consideremos sua aplicação ao problema do caixeiro viajante (TSP). No TSP, um conjunto de locais (por exemplo, cidades) e as distâncias entre eles são fornecidos.

O problema consiste em encontrar um roteiro fechado de duração mínima que visite cada cidade uma vez e apenas uma vez.

Para aplicar o ACO ao TSP, consideramos um gráfico definido pela associação do conjunto de cidades ao conjunto de vértices do gráfico. Este gráfico é chamado de gráfico de construção.

Como no TSP é possível mover-se de qualquer cidade para qualquer outra cidade, o grafo de construção está totalmente conectado e o número de vértices é igual ao número de cidades.

Definimos os comprimentos das arestas entre os vértices para serem proporcionais às distâncias entre as cidades representadas por esses vértices e associamos valores de feromônios e valores heurísticos com as arestas do gráfico. Os valores dos feromônios são modificados em tempo de execução e representam a experiência acumulada da colônia de formigas, enquanto os valores heurísticos são valores dependentes do problema que, no caso do TSP, são definidos como o inverso dos comprimentos das arestas. Cada formiga começa em uma cidade selecionada aleatoriamente (vértice do gráfico de construção). Então, a cada etapa da construção, ele se move ao longo das arestas do gráfico. Cada formiga guarda uma memória de seu caminho e, nas etapas subsequentes, escolhe entre as arestas que não conduzem aos vértices que já visitou. Uma formiga construiu uma solução depois de visitar todos os vértices do gráfico.

Em cada etapa da construção, uma formiga escolhe probabilisticamente a aresta a seguir entre aquelas que levam a vértices ainda não visitados. A regra probabilística é

influenciada por valores de feromônio e informações heurísticas: quanto maior o feromônio e o valor heurístico associado a uma aresta, maior a probabilidade de uma formiga escolher aquela aresta em particular.

Depois que todas as formigas completaram seu passeio, o feromônio nas arestas é atualizado. Cada um dos valores de feromônio é inicialmente diminuído em uma certa porcentagem. Cada aresta recebe então uma quantidade de feromônio adicional proporcional à qualidade das soluções a que pertence (há uma solução por formiga).

Este procedimento é aplicado repetidamente até que um critério de encerramento seja satisfeito.

### 3.2 Particle Swarm Optimization (PSO)

A otimização por enxame de partículas é uma técnica de otimização estocástica baseada em população para a solução de problemas de otimização contínua. É inspirado por comportamentos sociais em bandos de pássaros e cardumes de peixes.

Na otimização por enxame de partículas (PSO), um conjunto de agentes de software chamados partículas busca boas soluções para um determinado problema de otimização contínua. Cada partícula é uma solução do problema considerado e usa sua própria experiência e a experiência das partículas vizinhas para escolher como se mover no espaço de busca.

Na prática, na fase de inicialização, cada partícula recebe uma posição inicial aleatória e uma velocidade inicial. A posição da partícula representa uma solução do problema e, portanto, tem um valor, dado pela função objetivo. Enquanto se movem no espaço de busca, as partículas memorizam a posição da melhor solução que encontraram.

A cada iteração do algoritmo, cada partícula se move com uma velocidade que é uma soma ponderada de três componentes:

- (1) a velocidade anterior,
- (2) um componente de velocidade que impulsiona a partícula em direção ao local no espaço de busca onde ela encontrou anteriormente a melhor solução até agora, e
- (3) um componente de velocidade que impulsiona a partícula em direção ao local no espaço de busca onde as partículas vizinhas encontraram a melhor solução até agora.

### 3.2.1 Exemplo

O PSO tem como exemplo de uso a busca por um ponto máximo ou mínimo de uma função, com a vantagem do uso de centenas de partículas minúsculas, cada uma delas operando esta busca e comunicando às partículas vizinhas os seus resultados, o que acarreta em novos e melhores resultados conforme o número de gerações (tempo) vai aumentando.

A primeira aplicação prática do PSO foi no campo de treinamento de redes neurais e foi relatada junto com o próprio algoritmo (Kennedy e Eberhart 1995).

Muitas outras áreas de aplicação foram exploradas desde então, incluindo telecomunicações, controle, mineração de dados, design, otimização combinatória, sistemas de energia, processamento de sinal e muitos outros.

Até o momento, existem centenas de publicações relatando aplicações de algoritmos de otimização de enxame de partículas.

Embora o PSO tenha sido usado principalmente para resolver problemas de otimização único e irrestritos, algoritmos de PSO foram desenvolvidos para resolver problemas restritos, problemas de otimização multiobjetivo, problemas com paisagens que mudam dinamicamente e para encontrar soluções múltiplas.

### 3.3 Swarm Robotics

A robótica de enxame é o estudo de como projetar grupos de robôs que operam sem depender de nenhuma infraestrutura externa ou de qualquer forma de controle centralizado. Tem suas origens na inteligência de enxame e, de fato, pode ser definida como a inteligência de enxame aplicada. O foco principal da pesquisa em robótica por enxame era estudar e validar a pesquisa biológica.

A colaboração inicial entre roboticistas e biólogos ajudou a desenvolver pesquisas em robótica em enxame, que desde então se tornou um campo de pesquisa por direito próprio.

Nos últimos anos, o foco da robótica de enxame tem mudado: de um campo bioinspirado da robótica, a robótica de enxame está cada vez mais se tornando um campo de engenharia cujo foco está no desenvolvimento de ferramentas e métodos para resolver problemas reais.

O projeto de enxames de robôs é orientado pelos princípios da inteligência de enxames. Esses princípios promovem a realização de sistemas tolerantes a falhas, escaláveis e flexíveis.

A robótica de enxame promove o desenvolvimento de sistemas que são capazes de lidar bem com a falha de um ou mais de seus robôs constituintes: a perda de robôs individuais não implica na falha de todo o enxame. Essa tolerância a falhas é possibilitada pela alta redundância do enxame: o enxame não depende de nenhuma entidade de controle centralizado, líderes ou qualquer robô individual desempenhando uma função predefinida.

A técnica também permite o desenvolvimento de sistemas que são capazes de lidar bem com as mudanças no tamanho do seu grupo: idealmente, a introdução ou remoção de indivíduos não causa uma mudança drástica no desempenho do enxame.

A escalabilidade é habilitada por detecção e comunicação local: desde que a introdução e remoção de robôs não modifique drasticamente a densidade do enxame, cada robô individual continuará interagindo com aproximadamente o mesmo número de pares, aqueles que estão em seu alcance de detecção e comunicação.

Por fim, a robótica de enxame promove o desenvolvimento de sistemas capazes de lidar com um amplo espectro de ambientes e condições operacionais. A flexibilidade é possibilitada pela natureza distribuída e auto-organizada de um enxame de robôs: em um enxame, os robôs se alocam dinamicamente em diferentes tarefas para atender aos requisitos do ambiente específico e das condições operacionais; além disso, os robôs operam com base na detecção e comunicação local e não dependem de infraestrutura pré-existente ou de qualquer forma de informação global.

### **3.3.1 Exemplo**

O uso de robôs para lidar com tarefas perigosas é claramente atraente, pois elimina ou reduz os riscos para os humanos. A natureza perigosa dessas tarefas implica um alto risco de perda de robôs. Portanto, uma abordagem tolerante a falhas é necessária, tornando as tarefas perigosas um domínio de aplicação ideal para enxames de robôs.

Exemplos de tarefas perigosas que podem ser realizadas com enxames de robôs são desminagem, busca e resgate e limpeza de derramamentos tóxicos. As aplicações potenciais para enxames de robôs são aquelas em que é difícil ou mesmo impossível estimar com antecedência os recursos necessários para realizar a tarefa. Por exemplo, alocar recursos para gerenciar um vazamento de óleo pode ser muito difícil porque muitas vezes é difícil estimar a produção de petróleo e prever sua evolução temporal. Nesses casos, é necessária uma solução escalonável e flexível.

Um enxame de robôs pode ser uma solução atraente: robôs podem ser adicionados ou removidos a tempo de fornecer a quantidade adequada de recursos e atender aos requisitos da tarefa específica.

Exemplos de tarefas que podem exigir uma quantidade de recursos desconhecida a priori são busca e resgate, rastreamento e limpeza. Outro domínio de aplicação potencial para robótica de enxame são tarefas que devem ser realizadas em ambientes grandes ou não estruturados, nos quais não há infraestrutura disponível que possa ser usada para controlar os robôs - por exemplo, nenhuma rede de comunicação disponível ou sistema de localização global.

Enxames de robôs podem ser empregados para tais aplicações porque são capazes de agir de forma autônoma, sem a necessidade de qualquer infraestrutura ou qualquer forma de coordenação externa.

Exemplos de tarefas em ambientes não estruturados e grandes são exploração, vigilância, desminagem e busca e salvamento planetário subaquático ou extraterrestre. Alguns ambientes podem mudar rapidamente com o tempo. Por exemplo, em uma situação pós-terremoto, os edifícios podem desabar - alterando assim o layout do ambiente e criando novos perigos. Nestes casos, é necessário adotar soluções que sejam flexíveis e possam reagir rapidamente aos eventos.

A robótica de enxame pode ser usada para desenvolver sistemas flexíveis que podem se adaptar rapidamente a novas condições operacionais. Exemplos de tarefas em ambientes que mudam com o tempo são patrulhamento, recuperação de desastres e busca e resgate.

#### **4 CONSIDERAÇÕES FINAIS**

Sendo uma área consideravelmente recente, a Swarm Intelligence pode ser vista como forte alternativa para a solução de problemas de altíssima complexidade, com a utilização da escalabilidade fortemente presente em todas as soluções anteriormente citadas.

Para o problema matemático do caixeiro viajante (TSP), sempre foi considerado praticamente impossível de se obter uma solução ótima em larga escala, visto que a complexidade do mesmo aumentava consideravelmente com um número cada vez maior de cidades (vértices no grafo). No entanto, conforme vimos anteriormente, a utilização da Ant Colony Optimization (ACO) para este problema apresenta uma nova forma de busca pela solução, utilizando de uma meta-heurística baseada na natureza, e com custo computacional relativamente baixo, pois cada entidade (formiga) age de acordo com um nível reduzido de regras, e com o passar de algumas gerações, o melhor caminho é encontrado. De forma semelhante, o Particle Swarm Optimization (PSO) pode ser usado para resolver problemas de otimização de natureza mais complexa, simulando o movimento de um enxame de partículas.

Por fim conclui-se que entre os algoritmos e soluções citados ao longo do trabalho, a Swarm Robotic apresenta a abordagem mais promissora, principalmente quando diferentes atividades devem ser realizadas simultaneamente, alta redundância e a falta de um único ponto de falha são desejadas e também é tecnicamente inviável configurar a infraestrutura necessária para controlar os robôs em um sistema centralizado.

## REFERÊNCIAS

BALDASSARRE, G.; PARISI, D.; NOLFI, S. **Distributed coordination of simulated robots based on self-organization**. *Artificial Life*, 12(3):289–311, 2006.

BENI, G. **From swarm intelligence to swarm robotics**. In *Swarm Robotics*, LNCS 3342, pp. 1–9, 2005. Springer.

BONABEU, E.; DORIGO, M.; Theraulaz, G. **Swarm Intelligence: from natural to artificial systems**. New York: Oxford University Press. 1999.

KENNEDY, J.; EBERHART, R. **Particle Swarm Optimization**. *Proceedings of IEEE International Conference on Neural Networks*. IV. pp. 1942–1948. 1995.

SOARES, P. B.; CARNEIRO, T. C. J.; Calmon, J. L.; CASTRO, L. O. da C. de O. **Análise bibliométrica da produção científica brasileira sobre Tecnologia de Construção e Edificações na base de dados Web of Science**. *Ambiente Construído*, Porto Alegre, v. 16, n. 1, p. 175-185, jan./mar. 2016.

## **AGRADECIMENTOS**

Agradeço, primeiramente a Deus, pelos dons da vida e da razão.

A esta universidade, seu corpo docente e demais colaboradores que a mantiveram em pleno funcionamento, mesmo diante das mais diversas dificuldades.

Aos meus pais, por todo o apoio e incentivo prestados durante esta jornada.

E por último, mas não menos importante, à minha namorada Tainá, que todos os dias me faz ser uma pessoa melhor.